

On the computational complexity and generalization properties of multi-stage and recursive scenario programs

Nikolaos Kariotoglou, Kostas Margellos and John Lygeros

Abstract

We discuss the computational complexity and feasibility properties of scenario based techniques for uncertain optimization programs. We consider different solution alternatives ranging from the standard scenario approach to recursive variants, and compare feasibility as a function of the total computation burden. We identify trade-offs between the different methods depending on the problem structure and the desired probability of constraint satisfaction. Our motivation for this work stems from the applicability and complexity reduction when making decisions by means of recursive algorithms. We illustrate our results on an example from the area of approximate dynamic programming.

Index Terms

Scenario approach, randomized optimization, uncertain systems, approximate dynamic programming.

I. INTRODUCTION

Robust optimization comes up naturally in a range of problems from finance to robotics ([1], [2], [3]). Uncertain data is often present in the formulation of a decision making problem and the optimal solution is required to be robust against any possible uncertainty realization [4]. However, uncertainty may take values from an infinite and possibly unbounded set, which we might not know analytically, giving rise to a robust optimization problem that is in general not tractable [5], [6], [7]. A significant amount of research has concentrated on robust problems with structural characteristics and uncertainty sets of specific geometry for which robust decisions can be made by means of a tractable optimization program [8].

An alternative way to deal with data uncertainty is to formulate a chance constrained variant of the initial problem where the optimal decision is allowed to violate the robust constraint on a set of pre-specified measure. The authors in [4], [9] provide explicit solutions to such problems under assumptions on the probability distribution of the uncertainty. To avoid such assumptions one can make use of uncertainty samples (either based on historical data or via a scenario generation model) and construct decisions that satisfy the system constraints only for the sampled uncertainty scenarios. The feasibility and performance

N. Kariotoglou and J. Lygeros are with the Automatic Control Laboratory, Department of Information Technology and Electrical Engineering, ETH Zürich, Zürich 8092, Switzerland (e-mail: karioto@control.ee.ethz.ch; lygeros@control.ee.ethz.ch)

K. Margellos is with the Department of Industrial Engineering and Operations Research, UC Berkeley, Sutardja Dai Hall 330, Berkeley CA 94720, United States (e-mail: kostas.margellos@berkeley.edu)

The work of N. Kariotoglou was supported by the Swiss National Science Foundation under grant number 200021_137876.

properties of the solution can be generalized to quantify the confidence with which the optimizer of the scenario program satisfies the constraints for uncertainty realizations different than those used in the optimization process, providing a probabilistic link between scenario based and chance constrained optimization. The *scenario approach* introduced in [10], [11] can be used to provide such feasibility generalization statements for convex optimization problems. Beyond feasibility guarantees, [12], [13], [14] provide bounds on the amount of constraint violation and probabilistic performance. Generalization properties of similar nature can be obtained for non-convex optimization programs as well, using VC theoretic results [15], [16], [17]; the complexity, however, of the resulting solution depends on the so-called VC dimension (see [15] for a precise definition), which is in general difficult to compute.

Here we focus on the generalization properties of scenario based convex optimization problems using the scenario approach [10], [11], [18], [19]. The scenario approach deals with robust and chance constrained convex optimization problems by solving sampled programs constructed using a finite number of samples. The method provides bounds on the number of samples needed to provide guarantees about the feasibility of the optimal solution of the sampled program with respect to the original one. The number of required samples determines the total number of constraints that, together with the number of decision variables and the type of problem (linear program, quadratic program, second-order cone program, semi-definite program, etc.), determine the overall computation effort. The given bounds scale well with certain structural quantities of the underlying problem and with the design parameters, apply to any problem under relatively mild assumptions and can be shown to be tight in a specific class of problems [11], [20]. However, when one considers optimization problems with additional structure on the constraints, the generic bounds on the number of samples are not a sufficient performance measure. The same guarantees on the feasibility of a scenario based solution may be obtained by formulating several alternative scenario programs, each with a potentially different number of decision variables and constraints and hence different computational complexity. Here we investigate these trade-offs for a class of recursive optimization problems that naturally arises in applications such as stochastic model predictive control (SMPC) [21] and approximate dynamic programming (ADP) [22]. We consider two alternative structures, one with a single convex optimization problem with multiple constraint functions and one where the constraint functions are coupled. We show how, besides the standard scenario program, other types of scenario programs can be formulated for generic problems that respect these structures. These alternatives provide the same feasibility guarantees at potentially lower computation cost. We demonstrate this trade-off by benchmarking on a particular class of algorithms (primal-dual) and a particular class of problems (robust second-order cone problems). We also show how the stage-wise confidence and the violation level, typically treated as parameters in scenario programs, can be chosen by means of a convex optimization program to reduce the overall computation time. We demonstrate our results by applying them to a particular ADP algorithm developed for reachability problems.

Section II provides a statement of the problem under consideration. In Section III we present the different scenario based alternatives along with a pair of convex optimization problems that choose the stage-wise confidence and violation probability levels that result in the most favorable computational complexity. In Section IV we discuss the trade-off between the feasibility properties and the computational complexity of each alternative. Section V illustrates some features of the different algorithmic alternatives by means of a numerical example arising in ADP, while Section VI concludes the paper with some ideas on future research directions.

Notation: Let \mathbb{R} denote the real numbers, \mathbb{N} the natural numbers and \mathbb{N}_+ the positive natural numbers. In the derivations below all uncertainty samples are extracted from a (possibly unknown) uncertainty set Δ according to a fixed, possibly unknown probability measure \mathbb{P} . \mathbb{P}^S denotes the corresponding product measure for some $S \in \mathbb{N}_+$. We use i.i.d for identically and independently distributed uncertainty samples. Operator $|\cdot|$ denotes the cardinality of its argument, $\dim(A)$ denotes the dimension of a linear space A and $x \models y$ implies that x satisfies the statement in y .

II. CONVEX OPTIMIZATION PROGRAMS WITH MULTIPLE ROBUST CONSTRAINTS

Consider a compact convex set $\mathcal{X} \subseteq \mathbb{R}^d$, a possibly unbounded uncertainty set $\Delta \subseteq \mathbb{R}^w$, a convex cost function $f : \mathcal{X} \rightarrow \mathbb{R}$ and a set of $M \in \mathbb{N}_+$ convex constraint functions $g_i : \mathcal{X} \times \Delta \rightarrow \mathbb{R}$, $i = 1, \dots, M$; our results also extend to non-real valued (e.g. binary valued) uncertainties as long as they take values in a probability space. We are concerned with robust convex optimization problems (RCP) of the form:

$$\text{RCP} : \begin{cases} \min_{x \in \mathcal{X}} & f(x) \\ \text{s.t} & g_i(x, \delta) \leq 0, \forall \delta \in \Delta, \forall i \in \{1, \dots, M\}. \end{cases} \quad (1)$$

The set Δ may be infinite and possibly unbounded, rendering (1) a convex, semi-infinite optimization program. For such problems there is no general algorithm to obtain a solution, unless particular assumptions on the structure of Δ and the functions g_i are made (see for example [8], [23]).

A common approach to approximate the solution is to impose the constraints on a finite number of uncertainty instances. To this end, consider $S \in \mathbb{N}_+$ i.i.d samples $\{\delta^j\}_{j=1}^S$ extracted from Δ according to some, possibly unknown, underlying probability distribution, and a collection $\{\Delta_i\}_{i=1}^M$ of M subsets of $\{\delta^j\}_{j=1}^S$ such that for each $\delta \in \{\delta^j\}_{j=1}^S$ there exists i so that $\delta \in \Delta_i$, i.e. the sets may be overlapping but each δ belongs in at least one of them. The interpretation is that for each $i = 1, \dots, M$, the corresponding constraint $g_i(x, \delta)$ should be satisfied for all $\delta \in \Delta_i$, but not necessarily for all $\delta \in \Delta$. Problem (1) is then approximated by a scenario convex optimization program (SCP) of the form:

$$\text{SCP}[\Delta_1, \dots, \Delta_M] : \begin{cases} \min_{x \in \mathcal{X}} & f(x) \\ \text{s.t} & g_i(x, \delta) \leq 0, \forall \delta \in \Delta_i, \forall i \in \{1, \dots, M\}. \end{cases} \quad (2)$$

This is a convex optimization program with a finite number of decision variables and constraints, that can be solved to optimality by various numerical solvers (e.g. CPLEX, Gurobi, MOSEK). We impose the following assumption on $\text{SCP}[\Delta_1, \dots, \Delta_M]$:

Assumption 1. For any set $\{\delta^j\}_{j=1}^S$ and collection of subsets $\{\Delta_i\}_{i=1}^M$ with $S, M \in \mathbb{N}_+$, $\text{SCP}[\Delta_1, \dots, \Delta_M]$ is feasible, its feasibility region has a non-empty interior and its minimizer $x^*[\Delta_1, \dots, \Delta_M] : \Delta^S \rightarrow \mathcal{X}$ is unique.

We refer to [10], [20] for details on how the feasibility and uniqueness assumption can be relaxed; however, we keep these assumptions here to streamline the presentation of our results. Measurability of the minimizer $x^*[\Delta_1, \dots, \Delta_M]$ is assumed as needed, see [14], [24] for details.

Note that problem (2) and its minimizer are parametrized by the sets $\Delta_1, \dots, \Delta_M$. However, once the sets Δ_i are fixed, the unique (under Assumption 1) minimizer $x^*[\Delta_1, \dots, \Delta_M]$ of $\text{SCP}[\Delta_1, \dots, \Delta_M]$ is a mapping from Δ^S to \mathcal{X} and satisfies $g_i(x^*, \delta) \leq 0$, for all $\delta \in \Delta_i$ and $i \in \{1, \dots, M\}$. One of the challenges concerning this type of problem is to analyze the properties of x^* in terms of feasibility

(satisfiability of $g_i(x^*, \delta) \leq 0$ for all $\delta \in \Delta$) and performance (optimality of $f(x^*)$). Following the standard literature on the scenario approach [10], [11] we concentrate here on the feasibility properties of x^* as a function of the algorithm used to construct the solution; for a discussion on performance issues see [13], [14]. We establish that, depending on problem structure, there may be different ways of formulating the scenario program as a function of the choice of the number of samples S and the partition sets Δ_i . The computation effort necessary to solve the corresponding scenario programs differs, despite the fact that solutions have comparable feasibility properties. We will investigate the trade-offs between different design choices in the context of second order cone problems (SOCP) solved via primal-dual algorithms which are in general known to be of $\mathcal{O}\left((n+m)^3\right)$ complexity, where n denotes the dimension of the decision space and m the total number of constraints. Our motivation stems from the fact that a wide range of control-inspired optimization programs are SOCP, while primal-dual algorithms provide reliable termination and optimality conditions by iteratively reducing the duality gap. With straightforward modifications, related statements can be made for other classes of algorithms (e.g. gradient methods) and other classes of problems (linear programs, quadratic programs, semi-definite programs etc.).

III. FEASIBILITY PROPERTIES OF SCENARIO CONVEX PROGRAMS

We introduce four different approaches to formulate the scenario program: the standard scenario approach, the multi-stage scenario approach, the recursive scenario approach using the same samples at every recursive step and the recursive scenario approach using different samples at every recursive step. The standard scenario approach is the most general and applies to all problems in the form of RCP. The multi-stage and recursive counterparts assume particular structure on the constraint functions and exploit it to reduce computational complexity while maintaining similar feasibility properties. To streamline the comparison between different methods, we present here their main characteristics and devote the next section to discussing relative advantages.

A. The standard scenario approach

Let $\bar{\Delta} = \{\delta^j\}_{j=1}^S$ and assume that $\Delta_1 = \dots = \Delta_M = \bar{\Delta}$, in other words enforce each constraint for all elements in $\bar{\Delta}$. Denote by $\text{SCP}[\bar{\Delta}]$, $x^*[\bar{\Delta}]$ the resulting instance of $\text{SCP}[\Delta_1, \dots, \Delta_M]$ and its minimizer, respectively. For each (x, δ) , let $g(x, \delta) := \max_{i=1, \dots, M} g_i(x, \delta)$. The constraints $g_i(x, \delta) \leq 0$, $\forall \delta \in \bar{\Delta}$, $\forall i \in \{1, \dots, M\}$ are equivalent to $g(x, \delta) \leq 0$, $\forall \delta \in \bar{\Delta}$. Problem $\text{SCP}[\bar{\Delta}]$ was first studied in terms of feasibility in [10]. The following Theorem was then shown in [11].

Theorem 1 ([11], Theorem 2.4). *Choose $\varepsilon, \beta \in (0, 1)$ and fix $S \geq S(\varepsilon, \beta, d)$ where*

$$S(\varepsilon, \beta, d) := \min \left\{ N \in \mathbb{N} \mid \sum_{i=0}^{d-1} \binom{N}{i} \varepsilon^i (1-\varepsilon)^{N-i} \leq \beta \right\}. \quad (3)$$

Extract S samples i.i.d from Δ according to a probability measure \mathbb{P} , construct $\Delta_1 = \dots = \Delta_M = \bar{\Delta}$ and formulate $\text{SCP}[\bar{\Delta}]$. Under Assumption 1, the minimizer $x^[\bar{\Delta}]$ of $\text{SCP}[\bar{\Delta}]$ satisfies the chance constraint,*

$$\text{CCP}_\varepsilon : \mathbb{P}[g(x^*[\bar{\Delta}], \delta) > 0] \leq \varepsilon \quad (4)$$

with confidence (measured with respect to \mathbb{P}^S) at least $1 - \beta$.

Using the satisfiability notation “ \models ” (along the lines of [14]), the statement in Theorem 1 can be compactly written as $\mathbb{P}^S[x^*[\bar{\Delta}] \models \text{CCP}_\varepsilon] \geq 1 - \beta$. The interpretation of Theorem 1 is that, with certain confidence, the solution of the scenario convex program satisfies the robust constraint apart from a subset of the uncertainty space with measure at most ε . The computational complexity associated with constructing $x^*[\bar{\Delta}]$, along with the feasibility properties of Theorem 1, depend on the choice of ε, β and the number of decision variables d that implicitly affect the number of constraints (inspect (3)). The exact effect of each parameter is discussed in Section IV. Note that Theorem 1 remains unaffected if d is replaced by any upper bound on the number of the so-called support constraints (see [10] for a precise definition) other than the dimension of the decision space. Refinements along this direction are discussed in [18], [25], [26].

B. The multi-stage scenario approach

We impose here additional structure on the RCP by assuming that for any $\delta \in \Delta$ and for each $i = 1, \dots, M$, the constraint function $g_i(\cdot, \delta)$ depends on some (i.e. not necessarily all) of the decision variables. The set-up is then similar to the structure considered in [18], where the authors studied optimization programs with multiple chance constraints. For each $i = 1, \dots, M$, let $\mathcal{X}_i \subseteq \mathcal{X}$ denote the domain of each $g_i(\cdot, \delta)$ and $d_i = \dim(\mathcal{X}_i)$, where $\dim(\mathcal{X}_i)$ denotes the dimension of the smallest subspace of \mathbb{R}^d containing \mathcal{X}_i . We further assume that $d_i < d$ for at least one $i = 1, \dots, M$ to exclude the case where all constraint functions depend on all the decision variables; if this is not the case the subsequent analysis reduces to the standard scenario approach of Section III-A. We then have the following theorem due to [18], which serves as the multi-stage counterpart of Theorem 1.

Theorem 2 ([18], Theorem 4.1). *For each $i = 1, \dots, M$, choose $\varepsilon_i, \beta_i \in (0, 1)$, and fix $S_i \geq S(\varepsilon_i, \beta_i, d_i)$ where*

$$S(\varepsilon_i, \beta_i, d_i) := \min \left\{ N \in \mathbb{N} \mid \sum_{j=0}^{d_i-1} \binom{N}{j} \varepsilon_i^j (1 - \varepsilon_i)^{N-j} \leq \beta_i \right\}. \quad (5)$$

Extract $S = \sum_{i=1}^M S_i$ samples i.i.d from Δ according to a probability measure \mathbb{P} , construct $\{\Delta_i\}_{i=1}^M$ as in Section II with $|\Delta_i| = S_i$ and formulate $\text{SCP}[\Delta_1, \dots, \Delta_M]$. Under Assumption 1, for each $i = 1, \dots, M$, the minimizer $x^[\Delta_1, \dots, \Delta_M]$ of $\text{SCP}[\Delta_1, \dots, \Delta_M]$ satisfies the chance constraint,*

$$\text{CCP}_{\varepsilon_i} : \mathbb{P}[g_i(x^*[\Delta_1, \dots, \Delta_M], \delta) > 0] \leq \varepsilon_i, \quad (6)$$

with confidence (measured with respect to \mathbb{P}^{S_i}) at least $1 - \beta_i$.

As with Theorem 1, each d_i can be replaced by a tighter upper bound on the support constraints of g_i . Equation (6) in Theorem 2 establishes the feasibility properties of $x^*[\Delta_1, \dots, \Delta_M]$ for each separate constraint. However, no guarantees are provided on the probability that $x^*[\Delta_1, \dots, \Delta_M]$ satisfies all constraints simultaneously, i.e. CCP_ε in (4). This issue is addressed by the following corollary, that is a direct implication of Theorem 2.

Corollary 1. *Fix $\varepsilon, \beta \in (0, 1)$ and select $\varepsilon_i, \beta_i \in (0, 1)$, for all $i = 1, \dots, M$, such that $\sum_{i=1}^M \varepsilon_i = \varepsilon$ and $\sum_{i=1}^M \beta_i = \beta$. Under the set-up of Theorem 2 and Assumption 1 we have that $\mathbb{P}^S[x^*[\Delta_1, \dots, \Delta_M] \models \text{CCP}_\varepsilon] \geq 1 - \beta$, where CCP_ε is given in (4).*

Proof. The proof of Corollary 1 is essentially an application of the Boole-Bonferroni inequalities [27]. By Theorem 2 we have that $\mathbb{P}^S[x^*[\Delta_1, \dots, \Delta_M] \models \text{CCP}_{\varepsilon_i}] \geq 1 - \beta_i$, for all $i = 1, \dots, M$. By the subadditivity of \mathbb{P}_{Δ^S} we have that $\mathbb{P}^S[x^*[\Delta_1, \dots, \Delta_M] \models \text{CCP}_{\varepsilon_i}, \text{ for all } i = 1, \dots, M] \geq 1 - \sum_{i=1}^M \beta_i = 1 - \beta$. To complete the proof it suffices to show that $x^*[\Delta_1, \dots, \Delta_M] \models \text{CCP}_{\varepsilon_i}$ for all $i = 1, \dots, M$, implies that $x^*[\Delta_1, \dots, \Delta_M] \models \text{CCP}_{\varepsilon}$, where CCP_{ε} is given in (4). By the subadditivity of \mathbb{P} , and since $x^*[\Delta_1, \dots, \Delta_M] \models \text{CCP}_{\varepsilon_i}$ is equivalent to $\mathbb{P}[g_i(x^*[\Delta_1, \dots, \Delta_M], \delta) > 0] \leq \varepsilon_i$, we have that $\mathbb{P}[\exists i \in \{1, \dots, M\} \text{ such that } g_i(x^*[\Delta_1, \dots, \Delta_M], \delta) > 0] \leq \sum_{i=1}^M \varepsilon_i = \varepsilon$. Since by definition $g(x, \delta) := \max_{i=1, \dots, M} g_i(x, \delta)$, the last statement implies that $\mathbb{P}[g(x^*[\Delta_1, \dots, \Delta_M], \delta) > 0] \leq \varepsilon$, which is equivalent to $x^*[\Delta_1, \dots, \Delta_M] \models \text{CCP}_{\varepsilon}$ and concludes the proof. \square

The computational complexity associated with obtaining $x^*[\Delta_1, \dots, \Delta_M]$ with the feasibility properties of Corollary 1, depends on $\{d_i\}_{i=1}^M$ and the choices for $\{\varepsilon_i\}_{i=1}^M, \{\beta_i\}_{i=1}^M$. The obvious choice of $\varepsilon_i = \varepsilon/M$ and $\beta_i = \beta/M$ for $i = 1, \dots, M$ will in general be suboptimal; in Section III-E we formulate convex optimization problems to compute better choices.

C. Recursive scenario approach without re-sampling

In the sequel we consider RCP problems with specific structure on the constraint functions that enables us to tackle $\text{SCP}[\Delta_1, \dots, \Delta_M]$ in a sequential manner. We assume that the constraint functions $g_i(\cdot, \cdot, \cdot) : \mathcal{X}_i \times \mathcal{X}_{i+1} \times \Delta \rightarrow \mathbb{R}$ are pairwise coupled and convex with respect to their first argument, and $g_M(\cdot, \cdot) : \mathcal{X}_M \times \Delta \rightarrow \mathbb{R}$. As a consequence of this assumption we have by construction of RCP that $\mathcal{X} = \mathcal{X}_1 \times \dots \times \mathcal{X}_M$, which is a special case of the structure assumed in Section III-B. Let $x = (x_1, \dots, x_M)$ where $x_i \in \mathcal{X}_i$, for each $i = 1, \dots, M$. We further assume that the objective function is separable, i.e. $f(x) = \sum_{i=1}^M f_i(x_i)$. Such problem structures appear naturally in SMPC and ADP, as we demonstrate in Section V. The pairwise coupling structure can be relaxed to any form of stage-wise coupling as long as the constraint function at every stage is convex with respect to the decision variables.

The separable structure assumed, motivates the decomposition of $\text{SCP}[\Delta_1, \dots, \Delta_M]$ into a sequence of coupled scenario programs. For each $i = 1, \dots, M-1$ we define the following parametric scenario program

$$\text{SCP}_i[x_{i+1}, \Delta_i] : \begin{cases} \min_{x_i \in \mathcal{X}_i} & f_i(x_i) \\ \text{s.t} & g_i(x_i, x_{i+1}, \delta) \leq 0, \forall \delta \in \Delta_i \end{cases} \quad (7)$$

and $\text{SCP}_M[\Delta_M]$ analogously, with $g_M(x_M, \delta) \leq 0$ for all $\delta \in \Delta_M$, replacing the corresponding constraint in (7). We assume that all stage problems in (7) satisfy Assumption 1 for any fixed $x_{i+1} \in \mathcal{X}_{i+1}$; weaker assumptions are discussed in [28, Section 4].

Consider now the sequence of M pairwise coupled programs $\text{SCP}_i[x_{i+1}, \Delta_i]$ and let all sets Δ_i be identical, i.e. $\Delta_1 = \dots = \Delta_M = \bar{\Delta}$. Such optimization problems were referred to as cascading programs in [28], where the authors study the feasibility properties of a solution generated by sequentially solving a pair of coupled problems using the same set of uncertain scenarios. In particular, the following is a direct consequence of [28, Theorem 7].

Theorem 3 ([28], Theorem 7). *Let $d_i = \dim(\mathcal{X}_i)$ be the dimension of the smallest subspace of \mathbb{R}^d containing \mathcal{X}_i and $\bar{d} = \sum_{i=1}^M d_i$. Fix $\varepsilon, \beta \in (0, 1)$ and $S \geq S(\varepsilon, \beta, \bar{d})$, where $S(\varepsilon, \beta, \bar{d})$ is given by (3). Construct $x^* := (x_1^*, \dots, x_M^*)$, where each $x_i^*[\bar{\Delta}]$ is recursively computed from (7) with $\Delta_i = \bar{\Delta}$ for $i = 1, \dots, M$. We then have that $\mathbb{P}^S[x^*[\bar{\Delta}] \models \text{CCP}_{\varepsilon}] \geq 1 - \beta$.*

This recursive scenario based solution can be used to obtain feasibility properties for the solution of each step of the recursion. If $x^* := (x_1^*, \dots, x_M^*)$ is constructed according to Theorem 3, for any fixed $x_{i+1} \in \mathcal{X}_{i+1}$, with probability at least $1 - \beta_i$, $x_i^*[x_{i+1}, \bar{\Delta}]$ satisfies $\mathbb{P}[g_i(x_i^*[x_{i+1}, \bar{\Delta}], x_{i+1}, \delta) > 0] \leq \varepsilon_i$, for any $\varepsilon_i, \beta_i \in (0, 1)$ satisfying the equation $S(\varepsilon_i, \beta_i, d_i) \leq S$ where S is chosen such that $S \geq S(\varepsilon, \beta, \bar{d})$. In this case, however, the values of ε_i and β_i are not set a-priori and are not design choices; they are implicitly determined by the dimension of each subproblem. Consequently, the computational complexity of the recursive scenario approach only depends on \bar{d} and the choice of ε, β .

D. Recursive scenario approach with re-sampling

Consider the separable structure assumed in Section III-C and note that for a fixed $x_{i+1} \in \mathcal{X}_{i+1}$, $\text{SCP}_i[x_{i+1}, \Delta_i]$ is in the form of $\text{SCP}[\bar{\Delta}]$ considered in Section III-A. Fix ε_i and β_i and let the number of samples S_i , $i = 1, \dots, M$ be chosen according to (5). For any $x_{i+1} \in \mathcal{X}_{i+1}$, $\Delta_i \in \Delta^{S_i}$, let $x_i^*[x_{i+1}, \Delta_i] : \mathcal{X}_{i+1} \times \Delta^{S_i} \rightarrow \mathcal{X}_i$ be the minimizer of $\text{SCP}_i[x_{i+1}, \Delta_i]$. Theorem 1 implies that for all $i = 1, \dots, M - 1$, $x_i^*[x_{i+1}, \Delta_i]$ satisfies the chance constraint

$$\text{CCP}_{\varepsilon_i}[x_{i+1}] : \mathbb{P}[g_i(x_i^*[x_{i+1}, \Delta_i], x_{i+1}, \delta) > 0] \leq \varepsilon_i, \quad (8)$$

with probability at least $1 - \beta_i$, while for $i = M$, $x_M^*[\Delta_M]$ satisfies $\text{CCP}_{\varepsilon_M}$ with probability at least $1 - \beta_M$.

Using the parametrized scenario optimization problems in the form of (7) we can recursively construct a decision vector $x^* := (x_1^*, \dots, x_M^*)$, where for each $i = 1, \dots, M - 1$ the optimizer $x_i^*[x_{i+1}, \Delta_i]$ can be written as $x_i^*[\Delta_i, \dots, \Delta_M] : \Delta^{S_i} \times \dots \times \Delta^{S_M} \rightarrow \mathcal{X}_i$, satisfying

$$\mathbb{P}^{S_i} \left[x_i^*[\Delta_i, \dots, \Delta_M] \models \text{CCP}_{\varepsilon_i} \left[x_{i+1}^*[\Delta_{i+1}, \dots, \Delta_M] \right] \right] \geq 1 - \beta_i, \quad (9)$$

and for $i = M$, $\mathbb{P}^{S_M} [x_M^* \models \text{CCP}_{\varepsilon_M}] \geq 1 - \beta_M$. Note that due to the recursive process, x_i^* depends implicitly on all sets $\Delta_i, \dots, \Delta_M$. The following theorem can be used to compare the feasibility properties of a solution constructed in this way with a solution obtained using Theorems 1, 2 and 3.

Theorem 4. Fix $\varepsilon, \beta \in (0, 1)$ and select $\varepsilon_i, \beta_i \in (0, 1)$, for $i = 1, \dots, M$, such that $\sum_{i=1}^M \varepsilon_i = \varepsilon$ and $\sum_{i=1}^M \beta_i = \beta$. Construct $x^* := (x_1^*, \dots, x_M^*)$, where each $x_i^*[\Delta_i, \dots, \Delta_M]$ is recursively computed from (7), satisfying (9). Let $S = \sum_{i=1}^M S_i$ with $\{S_i\}_{i=1}^M$ chosen according to (5). We then have that $\mathbb{P}^S [x^*[\Delta_1, \dots, \Delta_M] \models \text{CCP}_{\varepsilon}] \geq 1 - \beta$.

Proof. Let $\bar{S}_i = \sum_{k=i}^M S_k$, $\bar{\varepsilon}_i = \sum_{k=i}^M \varepsilon_k$, $\bar{\beta}_i = \sum_{k=i}^M \beta_k$, $\bar{\Delta}_i = (\Delta_i, \dots, \Delta_M)$ and $\bar{x}_i^* = (x_i^*, \dots, x_M^*)$. We claim that for all $i = 1, \dots, M$, the following statement holds

$$\begin{aligned} & \mathbb{P}^{\bar{S}_i} \left[\mathbb{P} [g_M(x_M^*[\bar{\Delta}_M], \delta) > 0 \text{ or} \right. \\ & \left. \exists k \in \mathbb{N}_+, i \leq k < M : g_k(x_k^*[\bar{\Delta}_k], x_{k+1}^*[\bar{\Delta}_{k+1}], \delta) > 0] \leq \bar{\varepsilon}_i \right] \geq 1 - \bar{\beta}_i. \end{aligned} \quad (10)$$

The statement of the claim implies that, with confidence at least $1 - \bar{\beta}_i$, \bar{x}_i^* satisfies all constraints with indices greater than or equal to i , with probability at least $1 - \bar{\varepsilon}_i$.

If the claim holds, then for $i = 1$ we get the result. We show that the claim holds using induction. For $i = M$, (10) is trivially satisfied since $\bar{\Delta}_M = \Delta_M$ and $\text{SCP}_M[\Delta_M]$ is in the form of $\text{SCP}[\bar{\Delta}]$ considered

in Section III-A with $\bar{\Delta}, S, \varepsilon, \beta$ replaced by $\Delta_M, S_M, \varepsilon_M$ and β_M , respectively. Assume that (10) holds for some $1 < i < M$. By (9) we have that

$$\mathbb{P}^{S_{i-1}} \left[\mathbb{P} \left[g_{i-1}(x_{i-1}^*[\bar{\Delta}_{i-1}], x_i^*[\bar{\Delta}_i], \delta) > 0 \right] \leq \varepsilon_{i-1} \right] \geq 1 - \beta_{i-1}. \quad (11)$$

Using the fact that all samples are extracted independently, and (10), (11), hold for any uncertainty realization not in $\bar{\Delta}_i$ and $\bar{\Delta}_{i-1}$, respectively, (10), (11) would also hold with $\mathbb{P}^{\bar{S}_{i-1}}$ in place of \mathbb{P}^{S_i} and $\mathbb{P}^{S_{i-1}}$. From the resulting statements and the subadditivity of $\mathbb{P}^{\bar{S}_{i-1}}$ and \mathbb{P} , we can then show analogously to the proof of Corollary 1 that (10) holds with $i - 1$ in place of i . The latter implies that $\mathbb{P}^{\bar{S}_{i-1}}[x_{i-1}^* \models \text{CCP}_{\bar{\varepsilon}_{i-1}}] \geq 1 - \bar{\beta}_{i-1}$ and proves the claim. \square

The computational complexity associated with obtaining $x^*[\Delta_1, \dots, \Delta_M]$ with the feasibility properties of Theorem 4 depends on $\{d_i\}_{i=1}^M$ and the choices for $\{\varepsilon_i\}_{i=1}^M, \{\beta_i\}_{i=1}^M$. Notice that unlike the recursive scenario approach without re-sampling, $\{\varepsilon_i\}_{i=1}^M$ and $\{\beta_i\}_{i=1}^M$ are again design parameters for $i = 1, \dots, M$ and can be chosen in a way that reduces the computational complexity of the algorithm used to solve the corresponding optimization problems. We deal with this issue in the next section.

E. Complexity optimization

For the problems in Sections III-A and III-C, the number of decision variables d and $\sum_{i=1}^M d_i$ and the overall violation and confidence levels ε and β determine the total complexity. For Sections III-B and III-D on the other hand, although the overall violation and confidence are chosen a priori, the stage-wise levels $\{\varepsilon_i\}_{i=1}^M$ and $\{\beta_i\}_{i=1}^M$ are typically not fixed by the problem data and constitute a design choice that can affect the computational complexity due to the cubic dependence of SOCP solvers on the total number of samples and decision variables. Since the values of $\{d_i\}_{i=1}^M$ are fixed by problem data and generating samples from Δ can be hard, we focus on minimizing the total number of samples as an approximation to minimizing the total complexity. Throughout this section we replace the implicit sample size bound $S(\varepsilon_i, \beta_i, d_i) := \min \left\{ N \in \mathbb{N} \mid \sum_{j=0}^{d_i-1} \binom{N}{j} \varepsilon_i^j (1 - \varepsilon_i)^{N-j} \leq \beta_i \right\}$ that upper bounds the required sample size by the explicit bound $S(\varepsilon_i, \beta_i, d_i) \geq \frac{e}{e-1} \frac{1}{\varepsilon_i} \left(d_i - 1 + \ln \left(\frac{1}{\beta_i} \right) \right)$ due to [17]. For simplicity we treat the right-hand-side as an integer.

Proposition 1. *Consider the setup of Sections III-B and III-D where for each $i = 1, \dots, M$ the values of $d_i = \dim(\mathcal{X}_i)$ and $d = \dim(\mathcal{X})$ are fixed by the problem data. Fix $\varepsilon, \beta \in (0, 1)$. The problem of selecting $\{\varepsilon_i, \beta_i \in (0, 1)\}_{i=1}^M$ with $\sum_{i=1}^M \varepsilon_i \leq \varepsilon$, $\sum_{i=1}^M \beta_i \leq \beta$ that minimize the total number of samples $\sum_{i=1}^M S(\varepsilon_i, \beta_i, d_i)$, is a convex optimization program of the form:*

$$\begin{aligned} \min_{\{\varepsilon_i, \beta_i\}_{i=1}^M} \quad & \sum_{i=1}^M S(\varepsilon_i, \beta_i, d_i) \\ \text{subject to:} \quad & \sum_{i=1}^M \varepsilon_i \leq \varepsilon, \sum_{i=1}^M \beta_i \leq \beta, \varepsilon_i, \beta_i > 0, \forall i \in \{1, \dots, M\}. \end{aligned} \quad (12)$$

Proof. The function $S(\varepsilon_i, \beta_i, d_i)$ is convex with respect to ε_i, β_i since the Hessian matrix is positive definite for any $\varepsilon_i, \beta_i \in (0, 1)$. As a result, $\sum_{i=1}^M S(\varepsilon_i, \beta_i, d_i)$ is the sum of convex functions. \square

TABLE I
COMPLEXITY CHARACTERISTICS OF THE METHODS PRESENTED IN SECTIONS III-A- III-D.

	Section III-A	Section III-B	Section III-C	Section III-D
Number of problems	1	1	M	M
Samples per problem	$S \sim (3)$	$S_i \sim (5)$	$S \sim (3), d = \sum_{i=1}^M d_i$	$S_i \sim (5)$
Total number of samples	S	$\sum_{i=1}^M S_i$	S	$\sum_{i=1}^M S_i$
Decision variables per problem	$d = \dim(\mathcal{X})$	$d = \dim(\mathcal{X})$	$d_i = \dim(\mathcal{X}_i)$	$d_i = \dim(\mathcal{X}_i)$
Constraints per problem	MS	$\sum_{i=1}^M S_i$	S	S_i
Total complexity (SOCP)	$\mathcal{O}((d + MS)^3)$	$\mathcal{O}((d + \sum_{i=1}^M S_i)^3)$	$\sum_{i=1}^M \mathcal{O}((d_i + S)^3)$	$\sum_{i=1}^M \mathcal{O}((d_i + S_i)^3)$

The objective function of problem (12) is not in a standard form compatible with commercially available optimization software. As a result, one needs to implement a first or second order method to solve (12) (see for example [29]) taking advantage of the fact that both the gradient and Hessian matrix of the objective function are bounded with respect to ε_i, β_i in $[\mu, 1)$ for any $\mu > 0$. Fixing the confidence levels β_i a priori (e.g. $\beta_i = \beta/M$) simplifies the structure of (12) significantly and transforms the problem into a standard semi-definite program (SDP).

Proposition 2. Choose $\beta \in (0, 1)$ and fix the stage-wise confidence levels $\{\beta_i \in (0, 1)\}_{i=1}^M$ such that $\sum_{i=1}^M \beta_i \leq \beta$. Fix $\varepsilon \in (0, 1)$. For $c_i = \frac{e}{e-1} \left(d_i - 1 + \ln\left(\frac{1}{\beta_i}\right) \right)$, $i = 1, \dots, M$, the following SDP is equivalent to (12).

$$\begin{aligned}
 & \min_{\{t_i, \varepsilon_i\}_{i=1}^M} \sum_{i=1}^M t_i \\
 & \text{subject to: } \begin{bmatrix} t_i & \sqrt{c_i} \\ \sqrt{c_i} & \varepsilon_i \end{bmatrix} \succcurlyeq 0, \sum_{i=1}^M \varepsilon_i \leq \varepsilon, \varepsilon_i > 0, \forall i \in \{1, \dots, M\}
 \end{aligned} \tag{13}$$

Proof. The objective function in (12) can be written as $\sum_{i=1}^M c_i / \varepsilon_i$. Writing the problem in standard epigraph form and using Schur's complement we end up with the constraints in (13). \square

IV. DISCUSSION AND TRADE-OFFS

Each scenario based algorithm presented in Section III assumes a specific structure on the original RCP to construct a probabilistically feasible solution. Here we discuss differences between the feasibility properties of each solution and analyze the computational complexity of the associated algorithms as a function of design parameters.

A. Structure and feasibility properties

In contrast to the standard scenario approach of Section III-A, the multi-stage variant of Section III-B assumes that the domain of each constraint function in RCP is restricted to a subset of \mathcal{X} . By investigating each constraint separately, Theorem 2 provides guarantees on the probability that $x^*[\Delta_1, \dots, \Delta_M]$ satisfies every individual constraint, something that cannot be achieved with the standard scenario approach. In the recursive methodologies of Sections III-C and III-D we further restrict the structure of RCP by requiring the constraint functions to be pairwise coupled. In this way we relax the assumption regarding the convexity of the constraint functions. In particular, we require $g_i(x_i, x_{i+1}, \delta)$ to be convex with respect

to x_i , but do not require any convexity assumptions for the dependance on x_{i+1} . One situation where this can be of advantage is optimization programs with constraint functions that are bi-convex with respect to two decision vectors. Practically, such problems are often solved through a descent algorithm, alternating between optimizing with respect to one of the decision vectors while fixing the other decision vector to the value obtained at the preceding iteration. Theorem 3 allows us to provide probabilistic guarantees for the feasibility of the solution generated through such a descent algorithm, provided we fix a priori the number of iterations considered. Moreover, using the methodology of Section III-C which employs the same samples at every step of the recursive methodology ensures monotonicity of the objective function between consecutive steps of the recursion, that is crucial to ensure that the objective function decreases; see [28, Section 4].

Theorem 1, Corollary 1 and Theorems 3 and 4 all lead to a feasibility statement in the form of $\mathbb{P}^S[x^*[\Delta_1, \dots, \Delta_M] \models \text{CCP}_\varepsilon] \geq 1 - \beta$. Each method however requires a different number of samples to construct a solution and in turn the space on which the confidence related to the probability of constraint satisfaction is measured differs. In the standard scenario approach the total number of samples S is determined by the value of d and the choice of violation and confidence levels ε, β (inspect (3)). Assuming the same choice of ε and β , the total number of samples in the multi-stage scenario approach $\sum_{i=1}^M S_i$ can be greater or less than S depending on the values of $\{d_i\}_{i=1}^M$ (inspect (5) and the first two columns in Table I). In general, if each d_i is significantly smaller than d , then the total number of samples is smaller in the multi-stage scenario approach. The situation is analogous between the recursive scenario approach without and with re-sampling, where the total number of samples will be generally higher in the latter depending on the values of $\{d_i\}_{i=1}^M$ and the choices of $\{\varepsilon_i\}_{i=1}^M, \{\beta_i\}_{i=1}^M$ (see the last two columns in Table I). Note that for the multi-stage scenario approach and the recursive scenario approach with re-sampling we can use the methods of Section III-E to optimize over $\{\varepsilon_i\}_{i=1}^M$ and $\{\beta_i\}_{i=1}^M$ but there is no guarantee that this will lead to a smaller number of total samples since $\{d_i\}_{i=1}^M$ is fixed by problem data.

B. Complexity

Both the standard and multi-stage scenario approach of Sections III-A and III-B require solving a single problem of the same structure with the same number of decision variables but a potentially different number of constraints. The number of decision variables d is given by problem data, while the number of constraints depends on d , $\{d_i\}_{i=1}^M$ and the chosen ε, β and $\{\varepsilon_i\}_{i=1}^M, \{\beta_i\}_{i=1}^M$. In the standard scenario approach, we use the same samples S (see (3)) for each constraint function leading to a total of MS constraints. In the multi-stage scenario approach, we use different samples S_i (see (5)) for each constraint function leading to a total of $\sum_{i=1}^M S_i$ constraints, a number that can be minimized over $\{\varepsilon_i\}_{i=1}^M, \{\beta_i\}_{i=1}^M$ using the methods of Section III-E. The computational complexity of each method is reported in the first two columns of Table I. Depending on the ratio between the minimum value of $\sum_{i=1}^M S_i$ and MS , either of the two methods might be preferable.

The computational complexity of the recursive methodologies of Sections III-C and III-D depends on the number of decision variables and constraints per subproblem. Each subproblem involves a single constraint function and as a result the number of samples required by Theorems 3 and 4 coincides with the number of constraints. In the recursive scenario approach without re-sampling, we use the same number of samples S in every subproblem which depends on $\bar{d} = \sum_{i=1}^M d_i$ and the choice of ε, β (see

Theorem 3). If $\bar{d} = d$ (as is the case for example in some ADP problems, see Section V), the number of samples coincides with that of the standard scenario approach. In general however, it might very well be that $\bar{d} > d$ (as is the case, for example, in some SMPC problems). In the recursive scenario approach with re-sampling, the number of samples S_i in each subproblem coincides with the number of samples used in the multi-stage scenario approach and depends on d_i and the choice of ε_i, β_i (see Theorem 4). As in the multi-stage scenario approach, $\sum_{i=1}^M S_i$ can be minimized over $\{\varepsilon_i\}_{i=1}^M, \{\beta_i\}_{i=1}^M$ using the methods in Section III-E. The computational complexity of both recursive methods is reported in the last two columns of Table I. Whenever applicable, the recursive methods of Sections III-C and III-D can provide significant computational advantages, as illustrated in the next section.

V. NUMERICAL EXAMPLE: APPROXIMATE DYNAMIC PROGRAMMING

Dynamic programming (DP) recursions are widely used to characterize the value function of optimal control problems [30]. For systems with continuous states, explicitly computing the value function by space discretization methods suffers from the curse of dimensionality, making the process intractable for state spaces of even moderate dimensions. This has motivated the development of sophisticated ADP methods [31]. A recently established methodology is the linear programming approach to ADP [32] which projects the optimal value function on the span of a pre-selected set of basis functions, intersected with the feasibility region determined by a set of inequality constraints. The authors in [22], [33] developed an algorithm based on the linear programming approach to ADP, specifically to approximate the value function of stochastic reachability problems. In this section we use this algorithm to investigate the relative performance of the alternative scenario program formulations of Section III. We consider a simplified planar unicycle model with additive noise

$$\begin{bmatrix} \delta_1(i+1) \\ \delta_2(i+1) \end{bmatrix} = \begin{bmatrix} \delta_4(i) \cos(\delta_3(i)) + \delta_1(i) \\ \delta_4(i) \sin(\delta_3(i)) + \delta_2(i) \end{bmatrix} + w_i \quad (14)$$

where δ_1, δ_2 denote linear position, δ_3 yaw angle and δ_4 linear velocity. We assume that δ_3 and δ_4 are control inputs to the system and treat δ_1 and δ_2 as states. The noise terms $w_i \in \mathbb{R}^2$ are assumed to be independent for different i and identically distributed according to a multivariate normal distribution $\mathcal{N}(0, \Sigma)$ with diagonal covariance matrix. The combined state-action space is denoted by $\Delta = \Delta_x \times \Delta_u = \mathbb{R}^2 \times ([-0.5, 0.5] \times [-2\pi, 2\pi])$ where for $\delta = (\delta_1, \delta_2, \delta_3, \delta_4) = (\delta_x, \delta_u) \in \Delta$, δ_x corresponds to spatial coordinates while δ_u to control inputs. The symbol δ is used for the state and input variables since in the sequel we will be sampling from Δ . Given a target set $T = [0.8, 1]^2$, an avoid set $A = [-0.45, 0.25] \times [-0.2, 0.15]$ and a collection of time indexed safe sets $\{S_i\}_{i=1}^3 = \{[-1, 1]^2, [-0.3, 1]^2, [0.4, 1]^2\}$, the three step reach-avoid problem considered here is to maximize the probability that (14) reaches T while staying in the corresponding safe region $S_i \setminus A$ for time steps $i = 1, 2, 3$ (see Figure 1). The authors in [34] show that this reach-avoid problem can be solved via a DP recursion:

$$V_i^*(\delta_x) = \sup_{\delta_u \in \Delta_u} \underbrace{\{\mathbb{1}_T(\delta_x) + \mathbb{1}_{(S_i \setminus A) \setminus T}(\delta_x) \int_{\Delta_x} V_{i+1}^*(y) Q(dy|\delta)\}}_{h(\delta_x, \delta_u)} \quad (15)$$

$$V_4^*(\delta_x) = \mathbb{1}_T(\delta_x).$$

where V_i^* denotes the value function at stage i , Q denotes the transition kernel of the stochastic process in (14) and $\mathbb{1}_T, \mathbb{1}_{(S_i \setminus A) \setminus T}$ denote the indicator functions of the sets T and $(S_i \setminus A) \setminus T$ respectively.

We follow the ADP formulation for reach-avoid problems suggested in [32] and applied in [22], [33] to approximate (15). To this end we express the value function of each step in the DP recursion as a solution to an infinite dimensional linear program:

$$\begin{aligned} V_i^* \in \arg \inf_{V(\cdot) \in \mathcal{F}} \int_{\Delta_x} V(\delta_x) \nu(d\delta_x) \\ \text{subject to } V(\delta_x) \geq h(\delta_x, \delta_u), \forall \delta \in \Delta \end{aligned} \quad (16)$$

where ν is a (positive) measure supported on Δ_x and \mathcal{F} denotes the space of Borel-measurable functions in which, under mild assumptions, V_i^* resides [22]. Problems in the form of (16) are generally intractable and it is common in the literature to restrict the decision space to a finite dimensional subspace of \mathcal{F} to approximate each V_i^* . As suggested in [22], we restrict the decision space to a set of Gaussian radial basis functions (RBFs) with fixed centers and variances and use their span to approximate each V_i^* . Let $\{d_i\}_{i=1}^3 = \{200, 150, 100\}$ denote the cardinality of each basis set over the time horizon and $x = \{x_i\}_{i=1}^3$ with $x_i \in \mathbb{R}^{d_i}$, a collection of vectors corresponding to the weights of each RBF in the set. The reduction in the number of basis elements over the horizon is motivated by the reduction in the size of each safe set S_i . We denote by $L_i : \mathbb{R}^{d_i} \times \mathbb{R}^{d_{i+1}} \times \Delta \rightarrow \mathbb{R}$ the functions (linear in the first and second arguments) that for $i = 1, 2$ and each $\delta \in \Delta$ return the difference between the approximate value function at time i and the one-step-ahead reward at time $i + 1$ (observe the constraints in (16)). Each L_i implicitly depends on the safe, avoid and target regions at time i and the weights x_i, x_{i+1} completely determine its value over Δ . For $i = 3$, the function is defined as $L_3 : \mathbb{R}^{d_3} \times \Delta \rightarrow \mathbb{R}$ since the reach-avoid value function at $i = 4$ is known (15). Using this notation, the approximate reach-avoid value functions can be computed via a sequence of coupled robust linear programs:

$$\begin{aligned} \min_{x_i \in \mathbb{R}^{d_i}} x_i^\top I_i \\ \text{subject to: } L_i(x_i, x_{i+1}, \delta) \geq 0, \forall \delta \in \Delta \end{aligned} \quad (17)$$

where I_i denotes the element-wise integral over Δ of each RBF in the basis set with respect to the measure ν . The sequence of problems in (17) can be combined to a single problem as:

$$\begin{aligned} \min_{x \in \mathbb{R}^{d_1+d_2+d_3}} \sum_{i=1}^3 x_i^\top I_i \\ \text{subject to: } L_i(x_i, x_{i+1}, \delta) \geq 0, \forall \delta \in \Delta, i = 1, 2 \\ L_3(x_3, \delta) \geq 0, \forall \delta \in \Delta. \end{aligned} \quad (18)$$

Using the methods presented in Section III to solve (17) and (18) we can obtain an optimal solution for the weight vector x , possibly different for each method. Using the optimal weights, we can then directly construct the approximate value function of the stochastic reach-avoid problem for each $i = 1, 2, 3$.

We solved the problem with all methods and Table II compares the theoretical feasibility guarantees (column ε) with the empirical ones (column $\hat{\varepsilon}$) along with the associated complexities (columns “Sampling” and “Solver”). The empirical violation values were calculated by uniformly sampling 1000 realizations from Δ , other than those used in the optimization process, and computing the ratio between the number of realizations that resulted in constraint violation and 1000. We highlight with bold the parameters that can be chosen by the user and are not fixed by the problem data; for the multi-stage scenario approach and the recursive scenario approach with re-sampling, we have chosen the violation

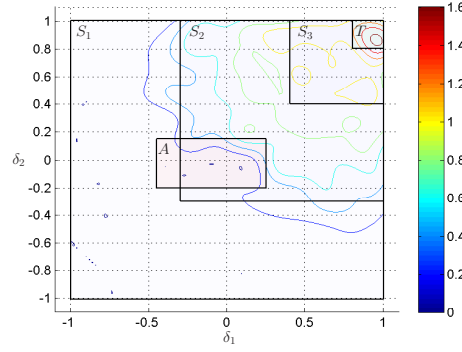


Fig. 1. Level sets of the approximate value function at $i = 1$ restricted on $[-1, 1]^2$, constructed using the method in Section III-D.

TABLE II
RESULTS OF ADP FOR REACH-AVOID USING THE METHODS PRESENTED IN SECTIONS III-A- III-D

	Horizon step	ε	$\hat{\varepsilon}$	$1 - \beta$	d	Constraints	Solver (sec)	Sampling (sec)
Section III-A	i=3	-	0.045	-	-	-	-	-
	i=2	-	0.034	-	-	-	-	-
	i=1	-	0.028	-	-	-	-	-
	Overall	0.1	0.038	0.97	450	43056	71	0.852
Section III-B	i=3	0.028	0.027	0.99	100	11865	-	0.8613
	i=2	0.034	0.019	0.99	150	14447	-	0.7786
	i=1	0.039	0.028	0.99	200	16632	-	0.7411
	Overall	0.1	0.065	0.97	450	42944	90	2.38
Section III-C	i=3	-	0.014	-	100	14352	1.78	-
	i=2	-	0.027	-	150	14352	4.7	-
	i=1	-	0.035	-	200	14352	8.82	-
	Overall	0.1	0.069	0.97	-	-	15.3	0.7671
Section III-D	i=3	0.028	0.009	0.99	100	11865	1.78	0.889
	i=2	0.034	0.016	0.99	150	14447	2.65	0.8242
	i=1	0.039	0.021	0.99	200	16632	4.76	0.6364
	Overall	0.1	0.045	0.97	-	-	9.19	2.35

levels ε_i at each stage by solving the complexity optimization program in (13). The associated confidence levels $1 - \beta_i, i = 1, 2, 3$ were all fixed to 0.99 to achieve an overall confidence $1 - \beta$ of at least 0.97. The basis centers and variances were sampled uniformly at random from each safe set and $(0, 0.01]$ respectively. All computations were done on an Intel Core i7 Q820 CPU clocked @1.73 GHz with 16GB of RAM memory, using the Gurobi optimization suite. Figure 1 shows the level sets of the approximation at time $i = 1$ restricted on $[-1, 1]^2$, constructed using the recursive scenario approach with re-sampling. Even though the optimal value function corresponds to a reach-avoid probability, the values of the approximation go above 1 since it is only an upper bound [22]. The accuracy of the approximation can be increased by increasing the number of basis elements or reducing the values of ε, β .

The results in Table II indicate that in this instance it is favorable to solve problems in a recursive manner since the same overall violation levels are respected while the computation times are smaller. Notice that in the standard scenario approach, the total number of samples is three times smaller than the total number of constraints since every sample is enforced on every constraint function in the horizon

separately. Moreover, in the multi-stage scenario approach and the recursive scenario approach with re-sampling we have to generate different samples for each constraint function in the horizon and thus sampling consumes more time. The reported solver times differ since differences in the sampled data affect solution time. In particular, the samples used for each of the constraints of the standard scenario approach are identical, giving structure to the problem which appears to be exploited by the solver. For the multi-stage scenario approach different samples are used for each constraint and the resulting optimization program has less structure. The differences in the reported sampling times (even when sample numbers are the same) are a consequence of the hit and run algorithm used to generate them [35]. The numbers reported are averaged over 10 runs of each method.

VI. CONCLUSION

We investigated the feasibility properties of different scenario based optimization programs, involving the standard scenario approach, its multi-stage counterpart as well as recursive variants that can be employed in case the problem exhibits a separable structure. We showed how confidence and violation levels can be treated as optimization assets and can be selected by means of convex optimization problems to reduce the computation time of the associated algorithm. We verified with a numerical example that the recursive structure often encountered in sequential decision making can be exploited, leading to much shorter computation times.

Our future work focuses on utilizing the insights gained in this paper in different problems where the assumed recursive structure is present. We already demonstrated the relevance and benefit of this in a class of approximate dynamic programming algorithms and believe that similar computational advantages will be observed in stochastic model predictive control problems. We also believe that recursive structures appear naturally in multi-agent systems where the decisions of one agent depend on the decision of another; in such cases using different samples between agents can have a significant impact on the required communication bandwidth. In terms of applications, we intend to use the recursive scenario approach discussed here to address surveillance tasks that are posed as reach-avoid problems [36].

REFERENCES

- [1] P. Kouvelis and G. Yu, *Robust discrete optimization and its applications*. Springer, 1997, vol. 14.
- [2] D. Bertsimas, D. B. Brown, and C. Caramanis, “Theory and applications of robust optimization,” *SIAM review*, vol. 53, no. 3, pp. 464–501, 2011.
- [3] A. Ben-Tal and A. Nemirovski, “Robust optimization—methodology and applications,” *Mathematical Programming*, vol. 92, no. 3, pp. 453–480, 2002.
- [4] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski, *Robust optimization*. Princeton University Press, 2009.
- [5] G. C. Calafiore, F. Dabbene, and R. Tempo, “Randomized algorithms for probabilistic robustness with real and complex structured uncertainty,” *IEEE Transactions on Automatic Control*, vol. 45, no. 12, pp. 2218–2235, 2000.
- [6] G. Calafiore, F. Dabbene, and R. Tempo, “A survey of randomized algorithms for control synthesis and performance verification,” *Journal of Complexity*, vol. 23, no. 3, pp. 301–316, 2007.
- [7] I. R. Petersen and R. Tempo, “Robust control of uncertain systems: Classical results and recent developments,” *Automatica*, vol. 50, no. 5, pp. 1315–1335, 2014.
- [8] D. Bertsimas and M. Sim, “The price of robustness,” *Operations research*, vol. 52, no. 1, pp. 35–53, 2004.
- [9] —, “Tractable approximations to robust conic optimization problems,” *Mathematical Programming*, vol. 107, no. 1-2, pp. 5–36, 2006.
- [10] G. C. Calafiore and M. C. Campi, “The scenario approach to robust control design,” *IEEE Transactions on Automatic Control*, vol. 51, no. 5, pp. 742–753, 2006.

- [11] M. C. Campi and S. Garatti, "The exact feasibility of randomized solutions of uncertain convex programs," *SIAM Journal on Optimization*, vol. 19, no. 3, pp. 1211–1230, 2008.
- [12] A. Lecchini, J. Lygeros, and J. Maciejowski, "Stochastic optimization on continuous domains with finite-time guarantees by Markov chain Monte Carlo methods," *IEEE Transactions on Automatic Control*, vol. 55, no. 12, pp. 2858–2863, 2010.
- [13] T. Kanamori and A. Takeda, "Worst-case violation of sampled convex programs for optimization with uncertainty," *Journal of Optimization Theory and Applications*, vol. 152, no. 1, pp. 171–197, 2012.
- [14] P. Mohajerin, T. Sutter, and J. Lygeros, "Performance Bounds for the Scenario Approach and an Extension to a Class of Non-convex Programs," *IEEE Transactions on Automatic Control (to appear)*, 2014. [Online]. Available: <http://arxiv.org/abs/1307.0345>
- [15] M. Vidyasagar, *A theory of learning and generalization*. Springer-Verlag New York, Inc., 2002.
- [16] R. Tempo, G. Calafiore, and F. Dabbene, *Algorithms for Analysis and Control of Uncertain Systems*. Springer, 2004.
- [17] T. Alamo, R. Tempo, and A. Luque, "On the sample complexity of probabilistic analysis and design methods," in *Perspectives in Mathematical System Theory, Control, and Signal Processing*. Springer, 2010, pp. 39–50.
- [18] G. Schildbach, L. Fagiano, and M. Morari, "Randomized Solutions to Convex Programs with Multiple Chance Constraints," *SIAM Journal on Optimization*, vol. 23, no. 4, pp. 2479–2501, Dec. 2013. [Online]. Available: <http://control.ee.ethz.ch/index.cgi?page=publications;action=details;id=4535>
- [19] G. C. Calafiore and D. Lyons, "Random convex programs for distributed multi-agent consensus," in *Control Conference (ECC), 2013 European*. IEEE, 2013, pp. 250–255.
- [20] G. C. Calafiore, "Random convex programs," *SIAM Journal on Optimization*, vol. 20, no. 6, pp. 3427–3464, 2010.
- [21] G. C. Calafiore and L. Fagiano, "Robust model predictive control via scenario optimization," *IEEE Transactions on Automatic Control*, vol. 58, no. 1, pp. 219–224, 2013.
- [22] N. Kariotoglou, T. Summers, M. Kamgarpour, and J. Lygeros, "A numerical approach to stochastic reach-avoid problems for markov decision processes," *arxiv*, 2014. [Online]. Available: <http://arxiv.org/abs/1411.5925>
- [23] A. Ben-Tal and A. Nemirovski, "Robust solutions of uncertain linear programs," *Operations research letters*, vol. 25, no. 1, pp. 1–13, 1999.
- [24] S. Grammatico, X. Zhang, K. Margellos, P. Goulart, and J. Lygeros, "A scenario approach for non-convex control design," *IEEE Transactions on Automatic Control (conditionally accepted)*, 2014.
- [25] G. Schildbach, L. Fagiano, C. Frei, and M. Morari, "The scenario approach for stochastic model predictive control with bounds on closed-loop constraint violations," *Automatica (to appear)*, 2014. [Online]. Available: <http://arxiv.org/pdf/1307.5640.pdf>
- [26] X. Zhang, S. Grammatico, G. Schildbach, P. Goulart, and J. Lygeros, "On the sample size of random convex programs with structured dependence on the uncertainty," *Automatica (submitted)*, 2014. [Online]. Available: <http://control.ee.ethz.ch/index.cgi?page=publications;action=details;id=4801>
- [27] A. Prékopa, *Stochastic programming*. Springer, 1995.
- [28] K. Margellos, M. Prandini, and J. Lygeros, "On the connection between compression learning and scenario based optimization," *IEEE Transactions on Automatic Control (conditionally accepted)*, 2014. [Online]. Available: <http://arxiv.org/abs/1403.0950>
- [29] Y. Nesterov, "Universal gradient methods for convex optimization problems," *Mathematical Programming*, pp. 1–24, 2013.
- [30] D. P. Bertsekas, *Dynamic programming and optimal control*. Athena Scientific Belmont, MA, 1995, vol. 1, no. 2.
- [31] W. B. Powell, *Approximate Dynamic Programming: Solving the curses of dimensionality*. John Wiley & Sons, 2007, vol. 703.
- [32] D. P. de Farias and B. Van Roy, "The linear programming approach to approximate dynamic programming," *Operations Research*, vol. 51, no. 6, pp. 850–865, 2003.
- [33] N. Kariotoglou, S. Summers, T. Summers, M. Kamgarpour, and J. Lygeros, "Approximate dynamic programming for stochastic reachability," in *Control Conference (ECC), 2013 European*. IEEE, 2013, pp. 584–589.
- [34] S. Summers and J. Lygeros, "Verification of discrete time stochastic hybrid systems: A stochastic reach-avoid decision problem," *Automatica*, vol. 46, no. 12, pp. 1951–1961, 2010.
- [35] D. P. Kroese, T. Taimre, and Z. I. Botev, *Handbook of Monte Carlo Methods*. John Wiley & Sons, 2011, vol. 706.
- [36] N. Kariotoglou, D. M. Raimondo, S. J. Summers, and J. Lygeros, "Multi-agent autonomous surveillance: a framework based on stochastic reachability and hierarchical task allocation," *Journal of dynamic systems, measurement, and control*, vol. 137, 2015.